



Agilent Technologies

ADS 2008
January 2008
Encrypted HSPICE Flow

Advanced Design System 2008

© Agilent Technologies, Inc. 2000-2008

395 Page Mill Road, Palo Alto, CA 94304 U.S.A.

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries. Microsoft®, Windows®, MS Windows®, Windows NT®, and MS-DOS® are U.S. registered trademarks of Microsoft Corporation. Pentium® is a U.S. registered trademark of Intel Corporation. PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated. UNIX® is a registered trademark of the Open Group. Java™ is a U.S. trademark of Sun Microsystems, Inc. SystemC® is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission. MATLAB® is a U.S. registered trademark of The Math Works, Inc.. HiSIM2 source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code in its entirety, is owned by Hiroshima University and STARC.

Errata The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

Warranty The material contained in this document is provided "as is", and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license. Portions of this product include the SystemC software licensed under Open Source terms, which are available for download at <http://systemc.org/>. This software is redistributed by Agilent. The Contributors of the SystemC software provide this software "as is" and offer no warranty of any kind, express or implied, including without limitation warranties or conditions or title and non-infringement, and implied warranties or conditions merchantability and fitness for a particular purpose. Contributors shall not be liable for any damages of any kind including without limitation direct, indirect, special, incidental and consequential damages, such as lost profits. Any provisions that differ from this disclaimer are offered by Agilent only.

Restricted Rights Legend If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Contents

- Encrypted HSPICE
 - ◦ Encrypted HSPICE Background
 - Licensing Requirements
 - - Signal Integrity Verification License
 - - HSPICE Simulation License
 - Following the Encrypted HSPICE Flow
 - - Importing an Encrypted HSPICE Subcircuit
 - - - ADS Design with Pin Interface and HSPICE INCLUDE Component
 - - - The HS_Include Component
 - Creating Your Design
 - Simulating the Design
 - Modifying HSPICE Kits for use in ADS

Encrypted HSPICE

The encrypted HSPICE flow involves using the HSPICE simulator with encrypted models for channel verification in the Advanced Design System (ADS) environment. This feature has been designed specifically for the high-speed, serial-link Signal Integrity designer whose primary flow is ADS. These designers typically develop transmitter and receiver models and would like to perform a final verification step using vendor supplied encrypted HSPICE models (see [Encrypted HSPICE Background](#)).

If you prefer to create your channel models and designs within the ADS framework using tools such as, Momentum, Broadband Spice Model Generator, EMDS for ADS, transmission line models, or measured data such as S-parameter files, then the encrypted HSPICE flow may be a good tool for you.

The encrypted HSPICE flow is a solution developed for time-domain channel verification using an HSPICE simulator in the ADS design flow. A subset of the most useful ADS passive models and sources are supported within the encrypted HSPICE flow and the feature is simple to use and setup in the ADS environment.

Note that this feature is not a generic solution and is targeted only for Signal Integrity verification. The feature inherits all of the limitations of the HSPICE simulator and will not work in a cosimulation environment; in particular, an ADS Ptolemy Cosimulation.

Encrypted HSPICE Background

Companies that distribute models that include their intellectual property (IP) generally deliver their models as either Input/Output Buffer Information Specification (IBIS) models, HSPICE models, or encrypted HSPICE models. IBIS

models tend to be faster; however, HSPICE models are generally considered to be more accurate. Therefore, supporting encrypted HSPICE models was an important aspect in the channel design/verification process. As an example, if a designer is using HSPICE models and the organization's IP is being delivered outside of the company, this designer is almost always required to use an encrypted HSPICE format. Because of this important need to protect the IP of various companies, the Agilent Technologies Signal Integrity solution has assumed that at least part of the circuit design could include components in the form of encrypted HSPICE models; therefore, driving the need for an encrypted HSPICE flow.

For more information on the Input/Output Buffer Information Specification, refer to the [IBIS Models](#) documentation.

Licensing Requirements

In addition to your standard ADS licenses, the following product licenses are required.

Signal Integrity Verification License

A signal integrity verification license is required to use the ADS tools for encrypted HSPICE flow.

ads_si_verification



Note

The ads_si_verification license is included by default in the majority of Signal Integrity and high-speed ADS bundles. You may want to check the availability of your license using your ADS license configuration tool before proceeding. For more information on ADS licensing, refer to Setting Up Licenses in your [Windows Installation](#) or [UNIX and Linux Installation](#) documentation.

HSPICE Simulation License

You must have the correct license for running the HSPICE simulator. The HSPICE Simulator and license are available from Synopsys, Inc. For more information on HSPICE licensing, contact Synopsys, Inc. or visit the Synopsys Web site at <http://www.synopsys.com/>.

Following the Encrypted HSPICE Flow

The encrypted HSPICE flow includes four main steps:

- Importing Encrypted HSPICE
- Creating Your Design

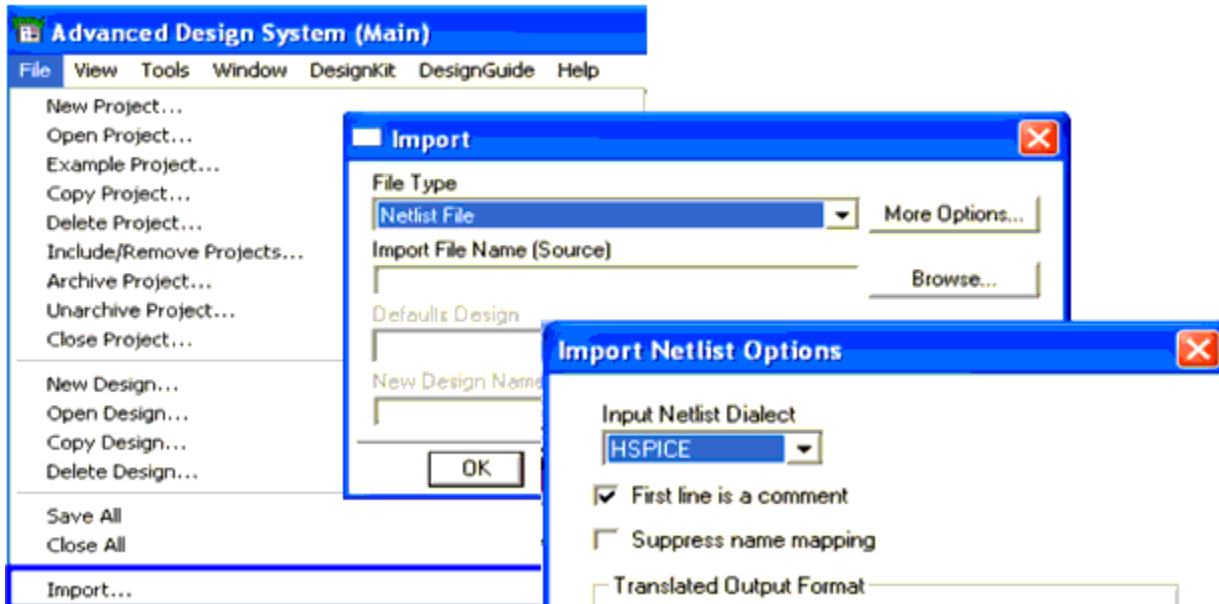
- Simulating the Design
- Evaluating Results

Importing an Encrypted HSPICE Subcircuit

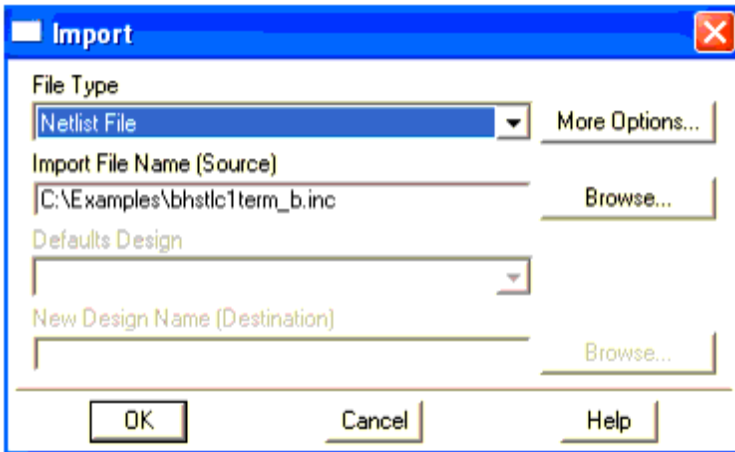
Importing an encrypted HSPICE subcircuit actually involves pointing to the encrypted HSPICE netlist fragment. A typical import provides an ADS representation of all of the files and then the external files are no longer needed. However, with the encrypted HSPICE flow, a placeholder is constructed that points to the actual file so that the location of the file can be passed off to the HSPICE simulator later.

To import your encrypted HSPICE netlist into an encrypted HSPICE subcircuit,

1. From the ADS Main Window, choose File > Import. The Import dialog box appears.
2. Click the More Options button to specify the netlist import options.
3. Using the Import Netlist Options dialog box, specify the following options:
4. Input Netlist Dialect: HSPICE
5. Translated Output Format: ADS Schematic

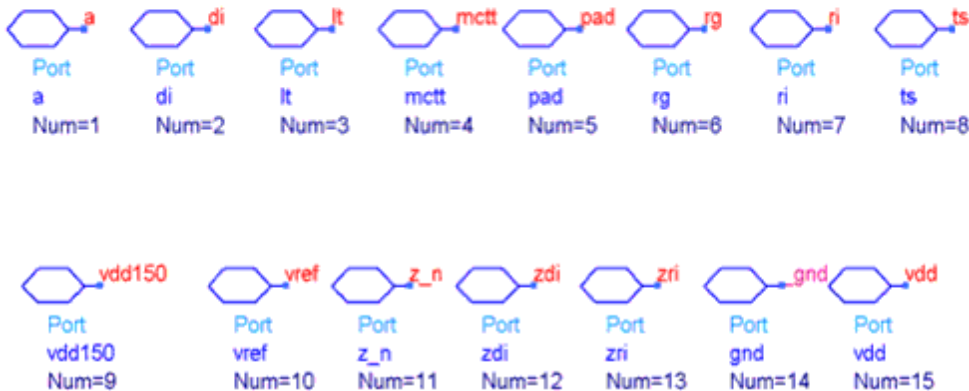
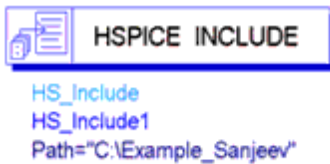


6. In the Import dialog box, specify the name and location of the encrypted HSPICE netlist and then click OK .



An ADS design will be created with a pin interface and an HS_INCLUDE component as shown in [ADS Design with Pin Interface and HSPICE INCLUDE Component](#). For more information on the HS_Include, refer to [The HS_Include Component](#).

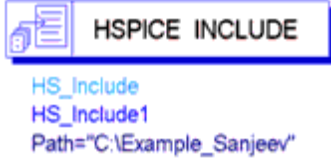
Note that the name of the new ADS design is the same name as the original encrypted HSPICE netlist with the exception of the .dsn extension.



ADS Design with Pin Interface and HSPICE INCLUDE Component

The HS_Include Component

The HSPICE INCLUDE (HS_Include) component is designed to instruct the simulator on how to use and apply a particular subcircuit as an encrypted HSPICE subcircuit. The component has only one parameter called, Path . The HSPICE INCLUDE component's Path parameter is used to provide the parent directory of where the encrypted HSPICE file can be found.



Note
The HSPICE _INCLUDE (HS_Include) component is automatically configured when you import an encrypted HSPICE netlist as described in [Importing an Encrypted HSPICE Subcircuit](#). The only reason you may want to modify the HSPICE_INCLUDE component's Path variable is if the original HSPICE source file is moved to a new location after you have performed the encrypted HSPICE netlist import.

To modify the path to the encrypted HSPICE netlist,

1. Double click the HSPICE INCLUDE component. The Incorporate HSPICE netlist dialog box appears.
2. In the Path field, modify the path to the parent directory of where the encrypted HSPICE file is located.
3. Click OK to change the path to the HSPICE INCLUDE component.

The encrypted HSPICE file must use the same name as the subcircuit that it contains with an added ".inc" extension. The ADS design, and thus the ADS subcircuit definition, must all have the same name as the encrypted HSPICE subcircuit.

Note that the HSPICE INCLUDE component is ignored by all ADS simulation controllers with the exception of the HSPICE TRANSIENT (HS_Tran) controller. For more information on this controller, refer to the [Encrypted HSPICE Simulation](#) documentation.

Note
Both the HSPICE INCLUDE (HS_Include) and HSPICE TRANSIENT (HS_Tran) controller are available under the Signal Integrity - Verification component palette.

Creating Your Design

The next step involves building your circuit using the imported encrypted HSPICE subcircuit (see [Importing an Encrypted HSPICE Subcircuit](#)). Note that the name of this subcircuit will be the same as the encrypted HSPICE netlist file.

Note
For the transient controller, use the HS_Tran controller. This is a different controller from the ADS transient controller. For more information, refer to the [Encrypted HSPICE Simulation](#) documentation.

Advanced Design System 2008

For a channel design, most of the ADS tools that you would normally use for creating a design are available. For example,

- Momentum
- Broadband Spice Model Generator
- EMDS for ADS
- Transmission line models, such as those documented in the Multilayer Interconnects section of the Distributed Components manual
- Use of measured data, such as S-parameter files
- Lumped Components
- Time Domain Sources

Simulating the Design

Before attempting to simulate your design, ensure that,

- You have the proper license available for running the HSPICE simulator. For more information, refer to [Licensing Requirements](#).
- You have hspice added to your \$PATH environment variable. If you need to add hspice to your path, you will need to restart ADS after making the change. The output from the HSPICE simulation is available to you as the simulation runs enabling you to monitor progress. Once the simulation is complete, a dataset is generated and the Data Display is open displaying the simulation results. This information can be used to evaluate your results.



Note

Simulation results returned from HSPICE will have all names lowercased.

Modifying HSPICE Kits for use in ADS

Some HSPICE kits must be modified into a form that ADS can use.

The example procedure below is provided to help you with modifying the Cyclone kit from Altera Corporation. The Altera HSPICE kit can be downloaded from: <http://www.altera.com/support/software/download/hspice/hsp-index.html>. The intent of this procedure is to package the encrypted portions of the HSPICE kit into modular subcircuits, which can then be used by the ADS netlist import and simulation processes. After downloading and unpacking the HSPICE kit:

1. Open the test netlist (e.g. Cyclone_1p5_8ma_inc.sp).
2. Find the subcircuit definition that wraps the encrypted content (CYCLONE_SINGLE_ENDED_IO, line 186).
3. Create a new file using the same name as the subcircuit, with an extension of ".inc" (cyclone_single_ended_io.inc).
4. Copy the ".subckt" to ".ends" lines (lines 186-192) to the new file. To comply with HSPICE conventions, make the first line of the file a comment (e.g. start it with "*").
5. Replace the ".include" line (line 6) with the contents of the indicated file (encrypted_cyclone_single_ended_io.inc).

Advanced Design System 2008

6. For any files that are referenced in ".include" lines in the test netlist (e.g. './models/Cyclong_typ_model.inc' and 'encrypted_c6851.inc'), copy those ".include" lines to the subcircuit file being created (cyclone_single_ended_io.inc). Insert the ".include" lines either just before the ".subckt" line or just after the ".ends" line.
7. Within an ADS schematic window,
 - Choose File > Import .
 - Select Netlist File as the File Type and then click More Options .
 - Select HSPICE from the Input Netlist Dialect drop-down list.
 - Then select the new ".inc" subcircuit that was created in steps 3 through 6 above (cyclone_single_ended_io.inc). The import process should create a PDE design file for the imported subcircuit (cyclone_single_ended_io.dsn).
8. In a new PDE design, place an instance of the subcircuit (cyclone_single_ended_io). Now design the channel, and place an HS_Tran controller to enable the encrypted HSPICE simulation.
9. When simulating, the HSPICE simulator reports errors indicating "no definition for rnnmd" and "no definition for rptmd". These are global variables that are referenced by the encrypted subcircuits. Edit the HS_Tran controller, and select the "Options" tab. In the "Global Parameters" section, define values for "rptmd" and "rnnmd". Save the new values and simulate again.

The netlist importer expects encrypted HSPICE files to be in exactly this format:

```
.subckt ...  
.param ...  
.prot ...  
.ends
```

That is, a single encrypted section inside of a subcircuit definition, with the associated pin names and parameter names and values. Any deviation from this - for example, unencrypted lines inside or outside of the subcircuit definition, multiple encrypted sections, etc. - will cause trouble for the netlist importer.